LOVELY PROFESSIONAL UNIVERSITY

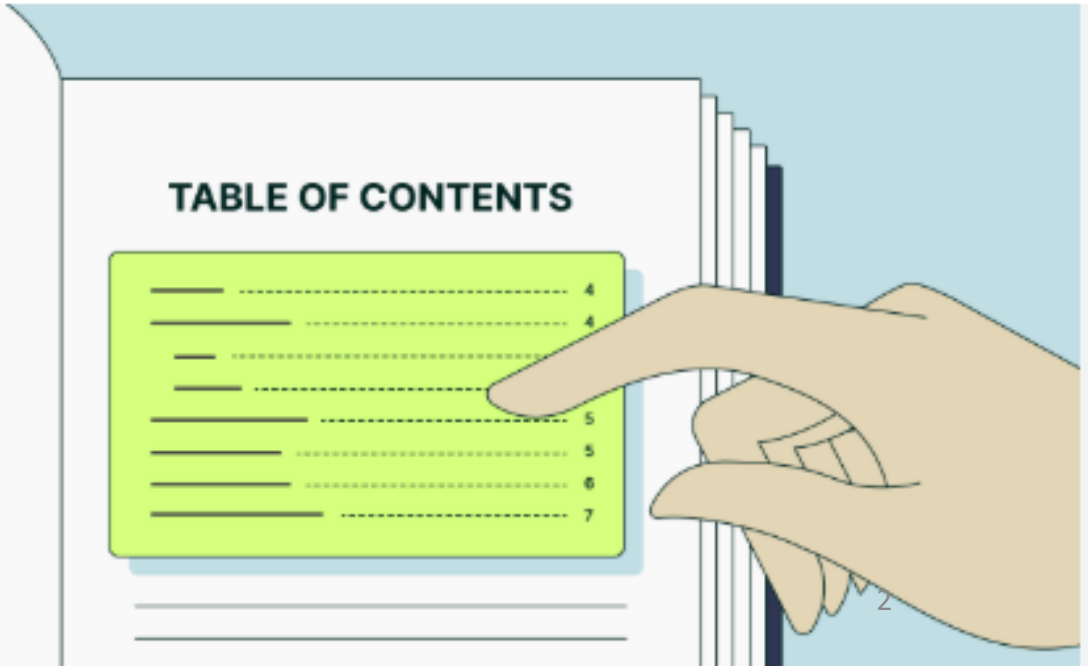
NAAC GRADE A++

Unit-2

S/w Design

Table of Content

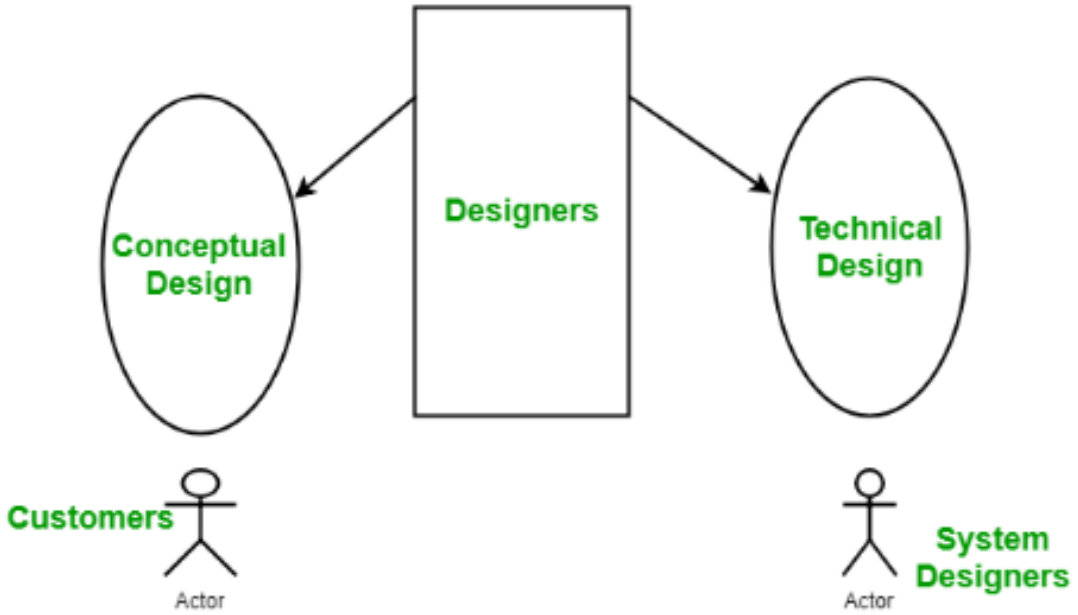
- Coupling
- Types of coupling



Design: A Two-Part Iterative Process

Conceptual Design

- which tells the customer <u>what the system will do.</u>



Technical Design

- which allows the system builders to understand the actual *<u>hardware and software needed to solve a customer's problem.</u>*

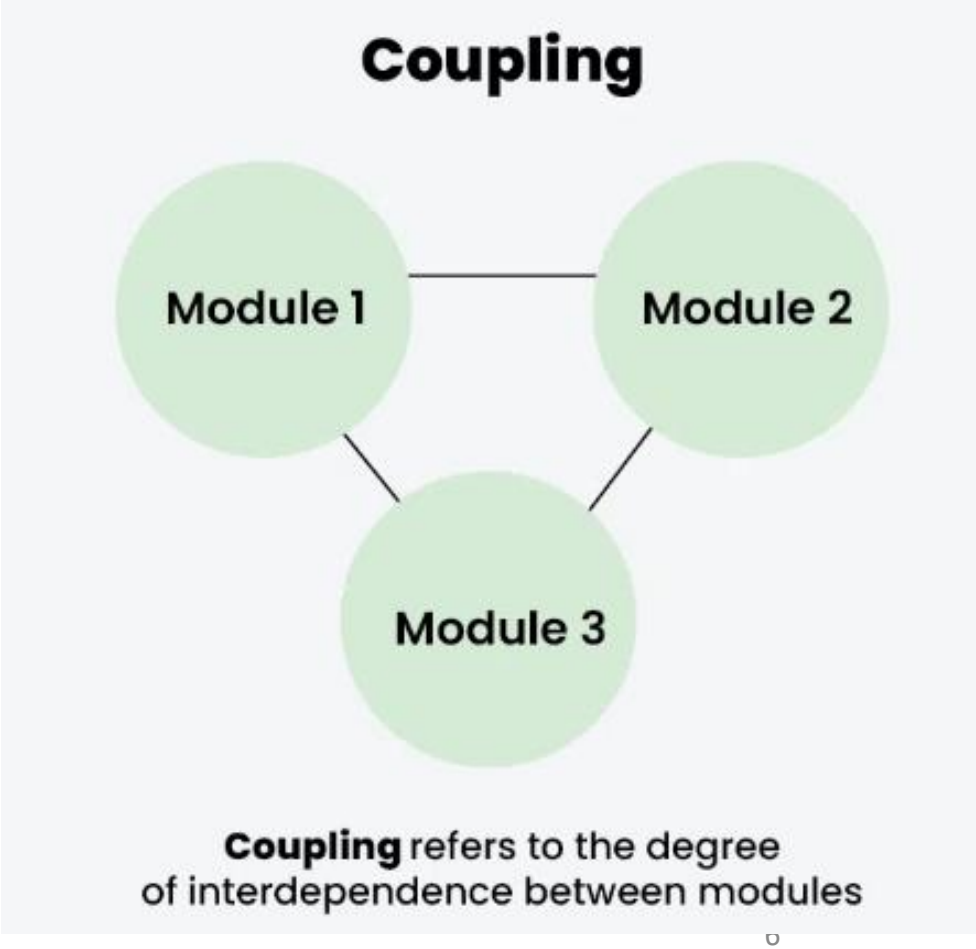
Characteristics of Good Design

- Component Independence
  - -High cohesion
  - -Low coupling
- Exception identification and handling
- Fault prevention and fault tolerance

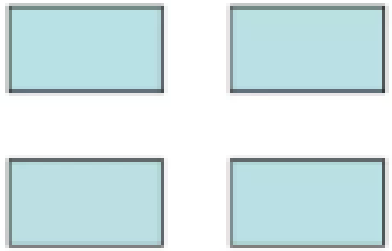
Coupling (inter module)

- Dependency between two or more modules.
- It measure the degree of interdependence between modules.
- A good software will have low coupling.

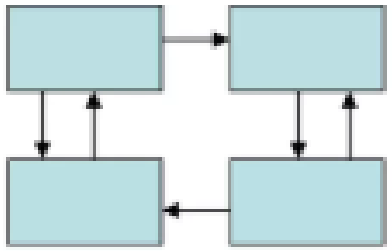
A functional independent module has minimal interaction with other modules



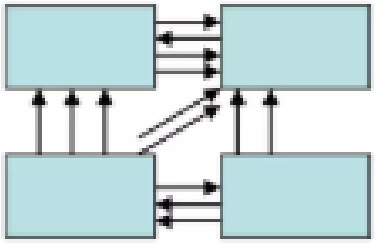
Coupling: Degree of dependence among components



No dependencies



Loosely coupled-some dependencies



Highly coupled-many dependencies

- High coupling makes modifying parts of the system difficult

E.g. modifying a component affects all the components to which the component is connected

Types of coupling

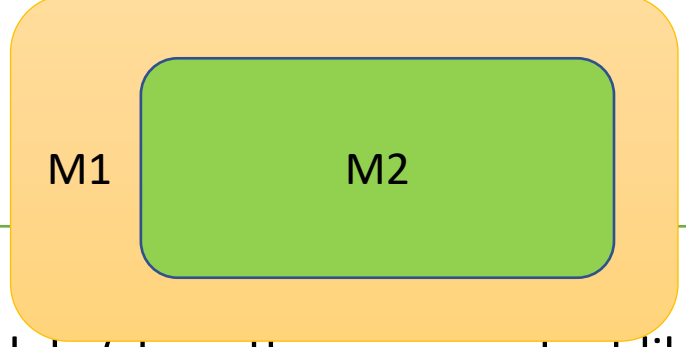
Dependency between two or more modules.

- Content Coupling
- Common Coupling
- External Coupling
- Control Coupling
- Stamp Coupling
- Data Coupling

bad/**Worst i.e. least desirable.**

best ……. A good software will have low coupling.

If modules are more inter related means they are highly interdependent to each other. So debugging or error isolation will become difficult in the particular module.



❑Content Coupling:

- When one module is a part or context of another module (share the same content like: functions, methods)
- Worst type of coupling

Example:

Two Apps:

- **Weather App:** Tracks and updates weather data.
- **Travel App:** Suggests travel plans based on the weather.

What Happens in Content Coupling?

- The **Travel App** directly accesses the **Weather App's database** to get temperature and conditions.

Why this is a Problem?

- **Tight Dependency:** Travel App relies on the structure of Weather App's database.
- Breaks Easily: If the Weather App changes temperature to temp_celsius, **the Travel App will crash**.

❑Common Coupling:

- Two modules are set to be common coupled **if they share some global data.**

Imagine multiple software modules (**like different apps**) sharing the same global data (like **a shared settings fil**e).

Example: A Game App and a Music App both *store user settings* (like **volume level**) in a shared file.

•If the Game App changes the volume setting to 80, the Music App will also get affected without knowing why.

Issues Caused:

•Unpredictable changes → *One module modifies data, affecting all others.*

•Hard to debug → If something breaks, it's difficult to find which module caused the issue.

•Tightly coupled → If the structure of shared data changes, all modules must be updated.

❑ External Coupling:

- **Two modules share an externally** import data format, communication protocols or device interface
- Communication to external tools and devices

Imagine you have a **smartphone app** that lets you print documents.

- The app sends a **print command** to a **printer** connected via Wi-Fi.
- If the **printer is turned off or disconnected**, the app **cannot function properly**.

The app depends on an **external hardware device** (the printer), creating **external coupling**.

Problems:

- **Device Failure:** If the printer is broken or offline, the app won't work.
- **Compatibility Issues:** If the printer model changes or a new printer brand is used, the app may need updates.
- **Network Dependency:** If the internet or Wi-Fi connection is lost, printing fails.

❑ Control Coupling:

- The modules is set to be controlled coupled if they communicate using control information with each other.

Example: Login System

Imagine a login system where **two functions are involved**

- **Function A**: Handles user input (username and password).
- **Function B**: Validates the login credentials.

Now, **Function A** receives the username and password, and based on the conditions:

- It sends the credentials to **Function B**.
- It also sends a **control flag**:
  - If the flag is "1", **Function B** will proceed with **login validation**.
  - If the flag is "0", **Function B** will **reset the password**.

❑ Stamp Coupling:

- When two modules communicate with each other by passing of data structure.

Consider a **book management system** with the following functions:

- **Function A**: Retrieves details of a book, including its **title**, **author**, **publication year**, and **ISBN number**.
- **Function B**: Needs to print only the **book title** and **author**.

- Now, **Function A** sends a **book object** (a data structure) to **Function B**, which contains all the information about the book, but **Function B** only needs the title and author to print them.

Passing unnecessary data creates a dependency on the entire structure, making the code more tightly coupled and harder to maintain.

❑ Data Coupling:

- occurs when two modules communicate by passing only the **necessary data** between them
- the modules communicate in a **clean and minimal** way

Example:

Consider a **product ordering system** with the following two functions:

- **Function A**: Calculates the price of a product.
- **Function B**: Applies a discount to the product price.

In this case:

- **Function A** calculates the price and passes only the **price value** to **Function B**.
- **Function B** applies a discount using the **price value** passed, without receiving unnecessary information like product name or description.

- **What does "coupling" refer to in software design?**

A) The level of dependency between modules
B) The process of writing modular code
C) The execution speed of a program
D) The number of functions in a module

- **Which type of coupling is the least desirable?**

A) Data Coupling
B) Control Coupling
C) Stamp Coupling
D) Content Coupling

- **In which coupling does one module control the behavior of another by passing flags or control parameters?**

A) Data Coupling
B) Stamp Coupling
C) Control Coupling
D) Common Coupling

- **What is the best type of coupling to aim for in software design?**

A) Data Coupling
B) Control Coupling
C) Common Coupling
D) Stamp Coupling

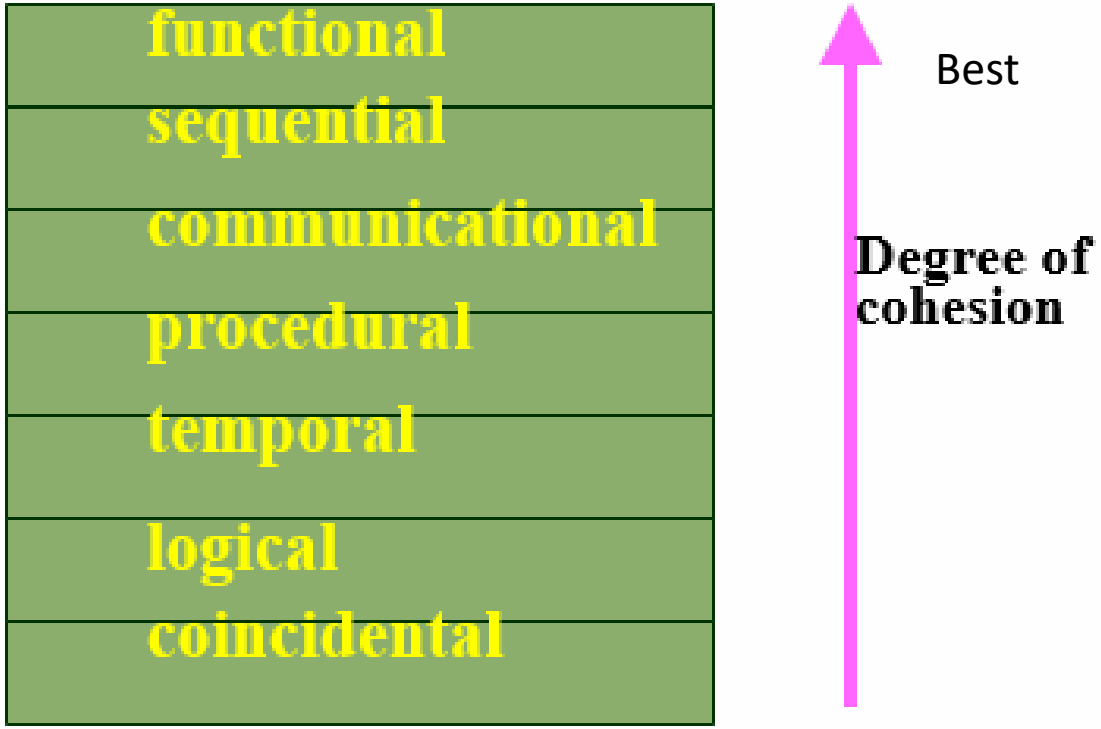
What is a disadvantage of tight coupling?

A) Easier maintenance
B) Higher reusability
C) Harder to modify and test modules independently
D) Increases flexibility

- **Which coupling occurs when two modules share the same global data?**

A) Data Coupling
B) Stamp Coupling
C) Common Coupling
D) Control Coupling

Types of cohesion



- **Cohesion (intra module)**

- Cohesion represent the *detail design (within module, how elements are interrelating to each other)*
- Cohesion means togetherness or group within the module.
- How we are grouping the various functions/ various methods inside the module.